



**Coutts, Fraser Kenneth and Thompson, Keith and Pestana, Jennifer and Proudler, Ian and Weiss, Stephan (2018) Enforcing eigenvector smoothness for a compact DFT-based polynomial eigenvalue decomposition. In: 10th IEEE Workshop on Sensor Array and Multichannel Signal Processing, 2018-07-08 - 2018-07-11. ,**

This version is available at <https://strathprints.strath.ac.uk/64015/>

**Strathprints** is designed to allow users to access the research output of the University of Strathclyde. Unless otherwise explicitly stated on the manuscript, Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Please check the manuscript for details of any other licences that may have been applied. You may not engage in further distribution of the material for any profitmaking activities or any commercial gain. You may freely distribute both the url (<https://strathprints.strath.ac.uk/>) and the content of this paper for research or private study, educational, or not-for-profit purposes without prior permission or charge.

Any correspondence concerning this service should be sent to the Strathprints administrator: [strathprints@strath.ac.uk](mailto:strathprints@strath.ac.uk)

The Strathprints institutional repository (<https://strathprints.strath.ac.uk>) is a digital archive of University of Strathclyde research outputs. It has been developed to disseminate open access research outputs, expose data about those outputs, and enable the management and persistent access to Strathclyde's intellectual output.

# Enforcing Eigenvector Smoothness for a Compact DFT-based Polynomial Eigenvalue Decomposition

Fraser K. Coutts\*, Keith Thompson\*, Jennifer Pestana†, Ian K. Proudler\*, Stephan Weiss\*

\* Department of Electronic & Electrical Engineering, University of Strathclyde, Glasgow, Scotland

† Department of Mathematics & Statistics, University of Strathclyde, Glasgow, Scotland

{fraser.coutts,stephan.weiss}@strath.ac.uk

**Abstract**—A variety of algorithms have been developed to compute an approximate polynomial matrix eigenvalue decomposition (PEVD). As an extension of the ordinary EVD to polynomial matrices, the PEVD will generate paraunitary matrices that diagonalise a parahermitian matrix. While iterative PEVD algorithms that compute a decomposition in the time domain have received a great deal of focus and algorithmic improvements in recent years, there has been less research in the field of frequency-based PEVD algorithms. Such algorithms have shown promise for the decomposition of problems of finite order, but the state-of-the-art requires a priori knowledge of the length of the polynomial matrices required in the decomposition. This paper presents a novel frequency-based PEVD algorithm which can compute an accurate decomposition without requiring this information. Also presented is a new metric for measuring a function's smoothness on the unit circle, which is utilised within the algorithm to maximise eigenvector smoothness for a compact decomposition, such that the polynomial eigenvectors have low order. We demonstrate through the use of simulations that the algorithm can achieve superior levels of decomposition accuracy to a state-of-the-art frequency-based method.

## I. INTRODUCTION

The expression of broadband multichannel problems via polynomial matrix representations [1] has been identified as a viable approach for broadband beamforming [2], [3], broadband angle of arrival estimation [4]–[6], and polyphase analysis and synthesis matrices for filter banks [7]. Parahermitian polynomial matrices, which are identical to their parahermitian conjugate, i.e.,  $\mathbf{R}(z) = \tilde{\mathbf{R}}(z) = \mathbf{R}^H(1/z^*)$  [7], are often central to these applications, and can arise as the  $z$ -transform of a space-time covariance matrix:  $\mathbf{R}(z) = \sum_{\tau} \mathbf{R}[\tau]z^{-\tau}$ .

A polynomial matrix eigenvalue decomposition (PEVD), which is defined in [8] as an extension of the eigenvalue decomposition to parahermitian matrices, uses finite impulse response (FIR) paraunitary matrices [9] to approximately diagonalise a space-time covariance matrix. Given an input parahermitian matrix  $\mathbf{R}(z) \in \mathbb{C}^{M \times M}$ , and its associated coefficient matrix  $\mathbf{R}[\tau]$ , PEVD algorithms generate an output diagonal matrix  $\mathbf{D}(z)$  containing eigenvalues, and a paraunitary matrix  $\mathbf{Q}(z)$  containing eigenvectors, such that

$$\mathbf{D}(z) \approx \tilde{\mathbf{Q}}(z)\mathbf{R}(z)\mathbf{Q}(z). \quad (1)$$

Equation (1) has only approximate equality, as the PEVD of a finite order polynomial matrix is generally not of finite order. Note that the decomposition in (1) is unique up to permutations and arbitrary allpass filters applied to the eigenvectors, provided that  $\mathbf{Q}(z)$  and  $\mathbf{D}(z)$  are selected analytic [10].

Existing PEVD algorithms include second-order sequential best rotation (SBR2) [8], sequential matrix diagonalisation (SMD) [11], and various evolutions of the algorithm families [12]–[14]. These algorithms employ iterative time domain schemes to approximately diagonalise a parahermitian matrix,

and typically encourage spectral majorisation [15] such that the resulting eigenvalues are ordered.

A DFT-based PEVD formulation, which transforms the problem into a pointwise-in-frequency standard matrix decomposition, is provided in [16]. The method can either return a spectrally majorised decomposition, or attempt to approximate maximally smooth eigenvalues. This algorithm has been shown to perform well for finite order problems [17], but requires an a priori guess of the length of the paraunitary matrix  $\mathbf{Q}(z)$  in the decomposition.

Here, we present a novel frequency-based PEVD algorithm which can compute an accurate PEVD without requiring an estimate of the paraunitary filter length. It has been demonstrated in [10] that the lowest order approximation to (1) is possible if we attempt to approximate analytic eigenvectors which — on the unit circle — are characterised by being infinitely differentiable. We utilise a cost function that measures the power in derivatives based on the Fourier coefficients of their discrete samples on the unit circle. This permits us to maximise eigenvector smoothness for a low order decomposition.

Below, Sec. II and Sec. III will introduce the smoothness metric and the proposed algorithm, respectively. A comparison of the algorithm's performance relative to the method in [16] is presented in Sec. IV, with conclusions drawn in Sec. V.

## II. SMOOTHNESS METRIC

A PEVD produces eigenvalues  $\lambda_m(e^{j\Omega})$  and eigenvectors  $\mathbf{q}_m(e^{j\Omega})$ , where  $\lambda_m(e^{j\Omega})$  is the  $m$ th diagonal element of  $\mathbf{D}(z)|_{z=e^{j\Omega}}$  and  $\mathbf{q}_m(e^{j\Omega})$  is the  $m$ th column of  $\mathbf{Q}(z)|_{z=e^{j\Omega}}$ . For the eigenvectors  $\mathbf{q}_m(e^{j\Omega})$  to be compact in the time domain, they must be maximally smooth on the unit circle [10]. In this section, we propose a novel metric to measure the smoothness of eigenvectors.

If  $K$  is sufficiently large and odd, the samples  $F_k = F(e^{j\Omega_k})$ ,  $\Omega_k = \frac{2\pi k}{K}$ ,  $k = 0 \dots (K-1)$ , of a function  $F(e^{j\Omega})$  permit its reconstruction via the Dirichlet kernel  $P(e^{j\Omega})$ :

$$F(e^{j\Omega}) = \sum_{k=0}^{K-1} F_k P(e^{j(\Omega - \Omega_k)}), \quad (2)$$

where

$$P(e^{j\Omega}) = \frac{1}{K} \cdot \frac{\sin(\frac{K}{2}\Omega)}{\sin(\frac{1}{2}\Omega)} = \frac{1}{K} \sum_{\ell=0}^{K-1} e^{-j\Omega(\ell - \frac{K-1}{2})}. \quad (3)$$

Thus,

$$F(e^{j\Omega}) = \frac{1}{K} \sum_{k=0}^{K-1} F_k \sum_{\ell=0}^{K-1} e^{-j(\Omega - \Omega_k)(\ell - \frac{K-1}{2})}. \quad (4)$$

For the  $p$ th derivative of  $F(e^{j\Omega})$ , we have

$$\frac{d^p}{d\Omega^p} F(e^{j\Omega}) = \frac{1}{K} \sum_{k=0}^{K-1} F_k \sum_{\ell=-(K-1)/2}^{(K-1)/2} (-j\ell)^p e^{-j(\Omega - \Omega_k)\ell}. \quad (5)$$

We are interested in the power contained within the  $p$ th derivative of  $F(e^{j\Omega})$ ,

$$\chi_p = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \frac{d^p}{d\Omega^p} F(e^{j\Omega}) \right|^2 d\Omega, \quad (6)$$

which provides some measure of the smoothness of  $F(e^{j\Omega})$ .

Note that due to orthogonality of the complex exponential terms and integration over an integer number of fundamental periods, for a Fourier series with arbitrary coefficients  $a_\ell$ ,

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \sum_{\ell} a_\ell e^{j\Omega\ell} \right|^2 d\Omega = \sum_{\ell} \frac{1}{2\pi} \int_{-\pi}^{\pi} |a_\ell e^{j\Omega\ell}|^2 d\Omega = \sum_{\ell} |a_\ell|^2. \quad (7)$$

Therefore, we can write

$$\chi_p = \frac{1}{K^2} \sum_{\ell=-(K-1)/2}^{(K-1)/2} \left| (-j\ell)^p \sum_{k=0}^{K-1} F_k e^{j\Omega_k \ell} \right|^2 \quad (8)$$

$$= \frac{1}{K^2} \sum_{\ell=-(K-1)/2}^{(K-1)/2} \ell^{2p} \mathbf{f}^H \mathbf{C}_\ell \mathbf{f} = \mathbf{f}^H \mathbf{C}^{(p)} \mathbf{f}, \quad (9)$$

where  $\mathbf{f} = [F_0, F_1, \dots, F_{K-1}]^T$  contains  $K$  frequency samples  $F_k = F(e^{j\Omega_k})$ ,  $k = 0 \dots (K-1)$ ,

$$\mathbf{C}_\ell = \begin{bmatrix} 1 & e^{j\frac{2\pi}{K}\ell} & \dots & e^{j\frac{2\pi}{K}(K-1)\ell} \\ e^{-j\frac{2\pi}{K}\ell} & 1 & & \vdots \\ \vdots & & \ddots & \vdots \\ e^{-j\frac{2\pi}{K}(K-1)\ell} & \dots & \dots & 1 \end{bmatrix}, \quad (10)$$

and

$$\mathbf{C}^{(p)} = \frac{1}{K^2} \sum_{\ell=-(K-1)/2}^{(K-1)/2} \ell^{2p} \mathbf{C}_\ell. \quad (11)$$

If we want to consider smoothness up to the  $P$ th derivative, we can form the metric

$$\begin{aligned} \chi^{(P)} &= \sum_{p=1}^P \chi_p = \mathbf{f}^H \left( \sum_{p=1}^P \mathbf{C}^{(p)} \right) \mathbf{f} \\ &= \mathbf{f}^H \left( \frac{1}{K^2} \sum_{\ell=-(K-1)/2}^{(K-1)/2} \mathbf{C}_\ell \sum_{p=1}^P \ell^{2p} \right) \mathbf{f} = \mathbf{f}^H \mathbf{C}_{(P)} \mathbf{f}, \end{aligned} \quad (12)$$

which calculates the sum of all powers of derivatives of  $\mathbf{f}$  up to and including the  $P$ th derivative. For even  $K$ , the derivation is similar to above and produces

$$\chi^{(P)} = \mathbf{f}^H \left( \frac{1}{K^2} \sum_{\ell=-\frac{K}{2}}^{\frac{K}{2}-1} \mathbf{C}_\ell \sum_{p=1}^P \ell^{2p} \right) \mathbf{f} = \mathbf{f}^H \mathbf{C}_{(P)} \mathbf{f}. \quad (13)$$

### III. PROPOSED ALGORITHM

#### A. Overview

The PEVD approach presented here uses a frequency-based scheme to obtain a solution to (1). In the discrete frequency domain, this equation has the form

$$\mathbf{R}[k] = \mathbf{Q}[k] \mathbf{D}[k] \mathbf{Q}^H[k], \quad k = 0, 1, \dots, K-1, \quad (14)$$

where  $\mathbf{Q}[k]$  contains eigenvectors,  $\mathbf{D}[k]$  contains eigenvalues, and  $\mathbf{R}[k]$  is obtained from the  $K$ -point DFT of  $\mathbf{R}[\tau]$ ,

$$\mathbf{R}[k] = \mathbf{R}(z)|_{z=w_K^k} = \sum_{\tau} \mathbf{R}[\tau] w_K^{k\tau}, \quad k = 0, 1, \dots, K-1,$$

where  $w_K = e^{-j2\pi/K}$ . An approximate PEVD is therefore obtained via  $K$  independent EVDs.

Noting that the decompositions in (1) and (14) are only unique up to permutations and phase shifts [10], an advantage of the frequency-based approach proposed here is the option to rearrange the eigenvalues and eigenvectors at each frequency

bin if desired. If the eigenvalues in  $\mathbf{D}[k]$  are arranged in descending order, approximate spectral majorisation of the resulting polynomial eigenvalues occurs. A smooth decomposition, whereby both eigenvalues and eigenvectors are continuous in frequency, typically produces a more compact decomposition, but requires additional effort. Sec. III-B discusses the method used to form a smooth decomposition.

Each eigenvector at each frequency bin can be influenced by an arbitrary scalar phase angle and still be valid. Discontinuities in phase between adjacent frequency bins can arise due to this ambiguity in eigenvector phase. When assessed via the metric of Sec. II, eigenvectors containing such phase discontinuities are not smooth and return high values of  $\chi^{(P)}$ . For a short paraunitary matrix  $\mathbf{Q}(z)$ , these discontinuities must be smoothed [10] and  $\chi^{(P)}$  decreased. This is achieved through the use of a phase alignment function, described in Sec. III-C, which uses Powell's 'dogleg' algorithm [18], [19] to maximise eigenvector smoothness.

Following the reordering (if desired) and phase alignment of  $\mathbf{Q}[k]$ ,  $\mathbf{Q}[\tau]$  is computed via the inverse DFT as

$$\mathbf{Q}[\tau] = \sum_{k=0}^{K-1} \mathbf{Q}[k] w_K^{-k\tau}, \quad \tau = 0, 1, \dots, K-1, \quad (15)$$

and  $\mathbf{D}[\tau]$  is found in a similar fashion.

#### B. Smooth Decomposition

In a smooth decomposition, the eigenvalues — and their eigenvectors — are arranged such that discontinuities between adjacent frequency bins are minimised. Such discontinuities can occur when the eigenvalues intersect at some frequencies. A relatively low complexity method for reordering the eigenvalues and eigenvectors from [16] is used in this algorithm, and is described below.

For a smooth decomposition, the eigenvectors in adjacent frequency bins are rearranged using the inner product

$$c_{ij}[k] = \mathbf{q}_i^H[k-1] \mathbf{q}_j[k], \quad (16)$$

where,  $\mathbf{q}_i[k]$  is the  $i$ th column of  $\mathbf{Q}[k]$ . For each eigenvector  $\mathbf{q}_i[k-1]$ ,  $i = 0 \dots M-1$ , a subsequent eigenvector  $\mathbf{q}_{i'}[k]$  is chosen from an initial set  $S = \{0 \dots M-1\}$  of the columns of  $\mathbf{Q}[k]$  such that

$$i' = \arg \max_{j \in S} \{|c_{ij}[k]|\}, \quad (17)$$

Once  $i'$  is identified, it is removed from the set:  $S = S - \{i'\}$ , and the next eigenvector is chosen. The selected eigenvectors are combined in a rearranged matrix  $\mathbf{Q}'[k] = [\mathbf{q}_{0'}[k] \dots \mathbf{q}_{(M-1)'}[k]]$ , and  $\mathbf{Q}[k]$  is set equal to  $\mathbf{Q}'[k]$ . The eigenvalues  $\mathbf{D}[k]$  are rearranged according to the reordering of the eigenvectors. This process is completed for  $k = 1 \dots K-1$ .

#### C. Phase Alignment

1) *Overview*: Phase alignment of eigenvectors in adjacent frequency bins is vital for a compact-order decomposition. Phase alignment can be achieved by finding the phase changes required for each eigenvector  $\mathbf{q}_m[k] \forall m, k$  to be maximally smooth according to metric  $\chi^{(P)}$ . The phase of the  $m$ th eigenvector at frequency bin  $k$  can be adjusted by an angle  $\theta_k$  according to  $\hat{\mathbf{q}}_m[k] = e^{j\theta_k} \mathbf{q}_m[k]$ . Below, we define an objective function which, when minimised via an unconstrained non-linear optimisation algorithm, can be used to determine the optimal  $\theta_k \forall k$  to maximise the  $m$ th eigenvector's smoothness.

2) *Objective Function Formulation*: The smoothness metric defined for odd  $K$  in (12) and even  $K$  in (13) is able

to measure the smoothness of a single function. We wish to compute a vector of phases  $\boldsymbol{\theta} = [\theta_0, \theta_1, \dots, \theta_{K-1}]^T \in \mathbb{R}^{K \times 1}$  such that all elements of the  $m$ th eigenvector  $\mathbf{q}_m[k] \forall k$  are maximally smooth. Our objective function therefore measures the smoothness of all elements of  $\mathbf{q}_m[k]$  and takes the form

$$f(\boldsymbol{\theta}) = \mathbb{R}\left\{\sum_{n=0}^{M-1} \mathbf{f}_n^H \mathbf{C}_{(P)} \mathbf{f}_n\right\} \quad (18)$$

where we choose  $\mathbf{f}_n^H = \mathbf{v}_n \text{diag}\{[e^{j\theta_0}, \dots, e^{j\theta_{K-1}}]\}$  and  $\mathbf{v}_n = [\mathbf{q}_{m,n}[0], \dots, \mathbf{q}_{m,n}[K-1]]$ , where  $\mathbf{q}_{m,n}[k]$  denotes the  $n$ th element (row) of eigenvector  $\mathbf{q}_m[k]$  at frequency bin  $k$ . Only the real component of the sum in the above equation is taken, as the imaginary component should be zero.

The above equation can be rearranged to form

$$f(\boldsymbol{\theta}) = \mathbb{R}\left\{\mathbf{u}^H \left(\sum_{n=0}^{M-1} \text{diag}\{\mathbf{v}_n\} \mathbf{C}_{(P)} \text{diag}\{\mathbf{v}_n^H\}\right) \mathbf{u}\right\}, \quad (19)$$

where  $\mathbf{u}^H = [e^{j\theta_0}, \dots, e^{j\theta_{K-1}}]$ .

Setting  $\Gamma = \sum_{n=0}^{M-1} \text{diag}\{\mathbf{v}_n\} \mathbf{C}_{(P)} \text{diag}\{\mathbf{v}_n^H\} \in \mathbb{C}^{K \times K}$  allows for the following compact representation:

$$f(\boldsymbol{\theta}) = \mathbb{R}\{\mathbf{u}^H \Gamma \mathbf{u}\}. \quad (20)$$

We therefore wish to obtain  $\boldsymbol{\theta}$  by solving

$$\boldsymbol{\theta} = \arg \min f(\boldsymbol{\theta}). \quad (21)$$

**3) Minimising the Objective Function:** Given its simplicity and relatively low cost [19], we employ Powell's iterative 'dogleg' trust region strategy [18] for the unconstrained minimisation of (20). In iteration  $i$ , a trust region strategy uses the gradient vector  $\mathbf{g}$  and approximate Hessian matrix  $\hat{\mathbf{H}}$  of the objective function to construct a second-order Taylor model:

$$m_i(\boldsymbol{\varphi}^{(i)}) = f(\boldsymbol{\theta}^{(i)}) + \boldsymbol{\varphi}^{(i)T} \mathbf{g} + \boldsymbol{\varphi}^{(i)T} \hat{\mathbf{H}} \boldsymbol{\varphi}^{(i)}. \quad (22)$$

This model approximates  $f(\boldsymbol{\theta}^{(i)} + \boldsymbol{\varphi}^{(i)})$  within a trusted region of radius  $R^{(i)}$  around the current point  $\boldsymbol{\theta}^{(i)}$  for some step  $\boldsymbol{\varphi}^{(i)}$ , which is identified according to

$$\boldsymbol{\varphi}^{(i)} = \arg \min_{\boldsymbol{\varphi}^{(i)}} m_i(\boldsymbol{\varphi}^{(i)}) \quad \|\boldsymbol{\varphi}^{(i)}\| < R^{(i)}. \quad (23)$$

The selected method first finds the minimiser of (22) along the steepest descent direction:

$$\boldsymbol{\varphi}_c^{(i)} = -\frac{\mathbf{g}^T \mathbf{g}}{\mathbf{g}^T \hat{\mathbf{H}} \mathbf{g}} \mathbf{g}. \quad (24)$$

If this point lies outside of the trusted region, the intersection of the line from the origin to  $\boldsymbol{\varphi}_c^{(i)}$  with the trust region boundary is used for  $\boldsymbol{\varphi}^{(i)}$ . Otherwise, the quasi-Newton point is found:

$$\boldsymbol{\varphi}_{qn}^{(i)} = -\hat{\mathbf{H}}^{-1} \mathbf{g}. \quad (25)$$

If this point lies within the trust region,  $\boldsymbol{\varphi}^{(i)} = \boldsymbol{\varphi}_{qn}^{(i)}$ , and the method moves to the next iteration. Otherwise, a solution is found at the intersection of the trust region boundary with the line from  $\boldsymbol{\varphi}_c^{(i)}$  to  $\boldsymbol{\varphi}_{qn}^{(i)}$ .

A parameter  $\rho$ , which compares the predicted reduction in the objective function with the actual reduction, is used to measure the strength of the approximation given by (22):

$$\rho = \frac{f(\boldsymbol{\theta}^{(i)}) - f(\boldsymbol{\theta}^{(i)} + \boldsymbol{\varphi}^{(i)})}{m_k(\mathbf{0}) - m_k(\boldsymbol{\varphi}^{(i)})}. \quad (26)$$

In practice, if  $\rho > 0.75$ , the approximation is good and the trust region is expanded. If  $\rho < 0.25$ , the approximation is poor and the trust region is decreased in size. Furthermore, if  $\rho > 0$ ,  $\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} + \boldsymbol{\varphi}^{(i)}$ ; otherwise,  $\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)}$ .

**4) Gradient Vector and Hessian Matrix Formulation:** The objective function in (20) can also be written as

$$f(\boldsymbol{\theta}) = \sum_{i=0}^{K-1} \sum_{k=0}^{K-1} u_i^* \gamma_{ik} u_k = \sum_{i=0}^{K-1} \sum_{k=0}^{K-1} \gamma_{ik} e^{j(\theta_i - \theta_k)}. \quad (27)$$

Taking the derivative with respect to  $\theta_\ell$ :

$$\begin{aligned} \frac{\partial f}{\partial \theta_\ell} &= -j \sum_{i=0, i \neq \ell}^{K-1} \gamma_{i\ell} e^{j(\theta_i - \theta_\ell)} + j \sum_{k=0, k \neq \ell}^{K-1} \gamma_{\ell k} e^{j(\theta_\ell - \theta_k)} \\ &= -j \sum_{i=0}^{K-1} \gamma_{i\ell} e^{j(\theta_i - \theta_\ell)} + j \sum_{k=0}^{K-1} \gamma_{k\ell}^* e^{-j(\theta_k - \theta_\ell)}. \end{aligned} \quad (28)$$

Note:  $\gamma_{\ell k} = \gamma_{k\ell}^*$  as  $\Gamma$  is Hermitian. Converting to matrix notation, the gradient vector is

$$\begin{aligned} \mathbf{g} &= \frac{\partial f}{\partial \boldsymbol{\theta}} = -j \text{diag}\{\mathbf{u}\} \Gamma^T \mathbf{u}^* + j \text{diag}\{\mathbf{u}^H\} \Gamma^H \mathbf{u} \\ &= 2\mathbb{R}\{j \text{diag}\{\mathbf{u}^H\} \Gamma^H \mathbf{u}\} = -2\mathbb{I}\{\text{diag}\{\mathbf{u}^H\} \Gamma \mathbf{u}\}. \end{aligned} \quad (29)$$

Taking the derivative of (28) with respect to  $\theta_p$ , with  $\theta_\ell$  now fixed, we get the equation for the off-diagonal elements of the Hessian matrix:

$$\begin{aligned} \frac{\partial^2 f}{\partial \theta_\ell \partial \theta_p} &= -j^2 \gamma_{p\ell} e^{j(\theta_p - \theta_\ell)} - j^2 \gamma_{p\ell}^* e^{-j(\theta_p - \theta_\ell)} \\ &= \gamma_{p\ell} e^{j(\theta_p - \theta_\ell)} + \gamma_{p\ell}^* e^{-j(\theta_p - \theta_\ell)}. \end{aligned} \quad (30)$$

So the off-diagonal (OD) terms of the Hessian matrix are:

$$\begin{aligned} \mathbf{H}_{\text{OD}} &= \text{diag}\{\mathbf{u}\} \Gamma^T \text{diag}\{\mathbf{u}^*\} + \text{diag}\{\mathbf{u}^H\} \Gamma^H \text{diag}\{\mathbf{u}\} \\ &= 2\mathbb{R}\{\text{diag}\{\mathbf{u}^H\} \Gamma \text{diag}\{\mathbf{u}\}\}. \end{aligned} \quad (31)$$

Taking the derivative of (28) with respect to  $\theta_\ell$ , we get the equation for the diagonal elements of the Hessian matrix:

$$\frac{\partial^2 f}{\partial \theta_\ell^2} = - \sum_{i=0, i \neq \ell}^{K-1} \gamma_{i\ell} e^{j(\theta_i - \theta_\ell)} - \sum_{k=0, k \neq \ell}^{K-1} \gamma_{\ell k} e^{j(\theta_\ell - \theta_k)}. \quad (32)$$

In matrix form, this equates to

$$\begin{aligned} \mathbf{H}_{\text{D}} &= \text{diag}\{-\text{diag}\{\mathbf{u}\} \hat{\Gamma}^T \mathbf{u}^* - \text{diag}\{\mathbf{u}^H\} \hat{\Gamma}^H \mathbf{u}\} \\ &= -2\mathbb{R}\{\text{diag}\{\text{diag}\{\mathbf{u}\} \hat{\Gamma}^T \mathbf{u}^*\}\}, \end{aligned} \quad (33)$$

where  $\hat{\Gamma} = \Gamma - \text{diag}\{\text{diag}\{\Gamma\}\}$  contains zeroes on its diagonal. The Hessian matrix is then given by

$$\begin{aligned} \mathbf{H} &= \mathbf{H}_{\text{OD}} - \text{diag}\{\text{diag}\{\mathbf{H}_{\text{OD}}\}\} + \mathbf{H}_{\text{D}} \\ &= 2\mathbb{R}\{\text{diag}\{\mathbf{u}^H\} \Gamma \text{diag}\{\mathbf{u}\}\} \\ &\quad - \text{diag}\{\text{diag}\{2\mathbb{R}\{\text{diag}\{\mathbf{u}^H\} \Gamma \text{diag}\{\mathbf{u}\}\}\}\} \\ &\quad - 2\mathbb{R}\{\text{diag}\{\text{diag}\{\mathbf{u}\} \hat{\Gamma}^T \mathbf{u}^*\}\} \\ &= 2\mathbb{R}\{\text{diag}\{\mathbf{u}^H\} \Gamma \text{diag}\{\mathbf{u}\}\} \\ &\quad - 2\mathbb{R}\{\text{diag}\{\text{diag}\{\Gamma\} + \text{diag}\{\mathbf{u}\} \hat{\Gamma}^T \mathbf{u}^*\}\} \\ &= 2\mathbb{R}\{\text{diag}\{\mathbf{u}^H\} \Gamma \text{diag}\{\mathbf{u}\}\} - 2\mathbb{R}\{\text{diag}\{\text{diag}\{\mathbf{u}\} \Gamma^T \mathbf{u}^*\}\}, \end{aligned} \quad (34)$$

since  $\text{diag}\{\text{diag}\{\mathbf{u}^H\} \Gamma \text{diag}\{\mathbf{u}\}\} = \text{diag}\{\Gamma\}$ .

The second part of (34) is typically less significant when compared to the first part; i.e.,

$$\frac{\|\mathbb{R}\{\text{diag}\{\text{diag}\{\mathbf{u}\} \Gamma^T \mathbf{u}^*\}\}\|_{\text{F}}}{\|\mathbb{R}\{\text{diag}\{\mathbf{u}^H\} \Gamma \text{diag}\{\mathbf{u}\}\}\|_{\text{F}}} \approx 10^{-3}, \quad (35)$$

where  $\|\cdot\|_{\text{F}}$  is the Frobenius norm. In numerical experiments, (34) is not always positive definite, which is a requirement for this method of minimisation [19]; however,



$2\Re\{\text{diag}\{\mathbf{u}^H\}\Gamma\text{diag}\{\mathbf{u}\}\}$  is positive semi-definite, as  $C_{(P)}$  and  $\Gamma$  are positive semi-definite. We propose the following approximation to the Hessian matrix, which is guaranteed to be positive definite and eliminates unnecessary computation:

$$\hat{\mathbf{H}} = 2\Re\{\text{diag}\{\mathbf{u}^H\}\Gamma\text{diag}\{\mathbf{u}\}\} + \alpha\mathbf{I}_K, \quad (36)$$

where  $\alpha$  is very small and  $\mathbf{I}_K$  is a  $K \times K$  identity matrix.

#### D. Algorithm Complexity

The complexity of the phase alignment step for a single eigenvector is  $\mathcal{O}(K^3)$  due to matrix inversion [16]; thus, the total complexity of the phase alignment step for  $M$  eigenvectors is of order  $\mathcal{O}(MK^3)$ . The computation of the frequency domain representation of  $\mathbf{R}[\tau] \forall \tau$ , the execution of  $K$  EVDs, and the formulation of a smooth decomposition are of lower complexity than this step; thus, the total complexity of the proposed algorithm is approximately  $\mathcal{O}(MK^3)$ .

### IV. ALGORITHM PERFORMANCE

To benchmark the proposed approach against the existing method in [16], this section first defines performance metrics for evaluating the performance of PEVD algorithms before setting out two simulation scenarios, over which an ensemble of simulations will be performed.

#### A. Performance Metrics

Denote the mean-squared reconstruction error for an approximate PEVD as

$$\text{MSE} = \frac{1}{M^2L'} \sum_{\tau} \|\mathbf{E}_R[\tau]\|_F^2, \quad (37)$$

where  $\mathbf{E}_R[\tau] = \hat{\mathbf{R}}[\tau] - \mathbf{R}[\tau] \forall \tau$ ,  $\hat{\mathbf{R}}(z) = \mathbf{Q}(z)\mathbf{D}(z)\tilde{\mathbf{Q}}(z)$ , and  $L'$  is the length of  $\mathbf{E}_R(z)$ . Furthermore, define the paraunitarity error as

$$\eta = \frac{1}{M} \sum_{\tau} \|\mathbf{E}_Q[\tau] - \mathbf{I}_M[\tau]\|_F^2, \quad (38)$$

where  $\mathbf{E}_Q(z) = \mathbf{Q}(z)\tilde{\mathbf{Q}}(z)$ ,  $\mathbf{I}_M[0]$  is an  $M \times M$  identity matrix, and  $\mathbf{I}_M[\tau]$  for  $\tau \neq 0$  is an  $M \times M$  matrix of zeroes. The output paraunitary matrix  $\mathbf{Q}(z)$  can be used in signal processing applications; a useful metric for gauging the implementation cost of this matrix is its length,  $L_Q$ .

#### B. Simulation Scenarios

The simulations below have been performed over an ensemble of  $10^3$  instantiations of  $\mathbf{R}(z) \in \mathbb{C}^{M \times M}$ ,  $M = 5$ , based on the randomised source model in [11]. In this source model, which generates ground truth  $\mathbf{D}(z)$  and  $\mathbf{Q}(z)$ , the order of  $\mathbf{D}(z)$  is 18 and the order of  $\mathbf{Q}(z)$  is 10, such that the total order of  $\mathbf{R}(z)$  is 38. The dynamic range of the eigenvalues is constrained to ensure that the average is around 50 dB.

The proposed algorithm considers smoothness up to the third derivative; i.e., we use the metric  $\chi^{(3)}$ . A maximum of 50 minimisation steps according to Sec. III-C3 are allowed, and a parameter of  $\alpha = 10^{-14}$  is used.

1) *Scenario 1*: The method in [16], henceforth referred to as the ‘existing’ algorithm, requires knowledge of  $L_Q$  prior to execution, and typically chooses a number of frequency bins  $K_e = 2L_Q + L - 2$ , where  $L$  is the length of the input matrix  $\mathbf{R}(z)$ . Our method, referred to as ‘proposed’, does not require an input value of  $L_Q$ , but instead outputs  $\mathbf{Q}(z)$  with length equal to the number of frequency bins  $K_p$  used in the decomposition. Note: outer lags of  $\mathbf{Q}[\tau]$  may contain negligible energy and can be trimmed using a threshold  $\mu$  in a process detailed in [8]. The first scenario provides the existing

TABLE I. AVERAGE MSE,  $\eta$ , AND A POSTERIORI  $L_Q$  COMPARISON.

Method	MSE	$\eta$	$L_Q$
existing <sup>1</sup> , $K_e = 47$	0.2139	0.03696	5
existing <sup>1</sup> , $K_e = 57$	$6.923 \times 10^{-6}$	$2.319 \times 10^{-6}$	10
existing <sup>2</sup> , $K_e = 131$	$1.251 \times 10^{-9}$	$8.596 \times 10^{-10}$	47
existing <sup>2</sup> , $K_e = 151$	$5.854 \times 10^{-9}$	$3.296 \times 10^{-9}$	57
proposed <sup>1,2</sup> , $K_p = 47$	$9.648 \times 10^{-18}$	$1.011 \times 10^{-15}$	47
proposed <sup>1,2</sup> , $K_p = 57$	$1.197 \times 10^{-22}$	$6.179 \times 10^{-19}$	57
proposed <sup>1,2,\ddagger</sup> , $K_p = 47$	$2.341 \times 10^{-10}$	$8.116 \times 10^{-11}$	23.35
proposed <sup>1,2,\ddagger</sup> , $K_p = 57$	$3.278 \times 10^{-10}$	$8.211 \times 10^{-11}$	23.22

<sup>1</sup> Simulation scenario 1

<sup>2</sup> Simulation scenario 2

<sup>\ddagger</sup>  $\mathbf{Q}[\tau]$  truncated using threshold  $\mu = 10^{-10}$  and scheme from [8]

method with values of  $L_Q \in \{5, 10\}$ , and uses the same generated values of  $K_e = K_p \in \{47, 57\}$  for both algorithms. Note that each algorithm is instructed to produce a smooth decomposition.

2) *Scenario 2*: To test the case where both algorithms produce  $\mathbf{Q}(z)$  of the same length, the second scenario provides the proposed method with values of  $K_p \in \{47, 57\}$ , and provides the existing method with  $L_Q = K_p$  such that  $K_e \in \{131, 151\}$ .

#### C. Results

The ensemble-averaged mean squared reconstruction error,  $\eta$ , and  $L_Q$  were calculated for both algorithms for both simulation scenarios, and can be seen in Tab. I. The table demonstrates that the proposed approach is able to provide extremely low decomposition MSE and paraunitarity error. Furthermore, the existing method is not capable of achieving such performance even when using significantly more frequency bins and generating paraunitary filters of the same length. The algorithmic complexity of both algorithms is approximately  $\mathcal{O}(MK^3)$  [16], [17]; thus the choice of DFT length  $K$  is extremely significant. Also shown is the impact of truncation on performance: if paraunitary filter length is of critical importance, then MSE and  $\eta$  can be sacrificed to generate shorter filters. Typically, increasing  $K$  is shown to improve MSE and  $\eta$  at the expense of higher  $L_Q$ .

### V. CONCLUSION

In this paper, we have introduced a novel frequency-based algorithm capable of computing a compact PEVD. This algorithm makes use of a newly developed metric for measuring the smoothness of a function on the unit circle. By minimising this metric for the eigenvectors produced by the algorithm, we have successfully modified the phase responses of the eigenvectors and enforced their compactness in the time domain.

Simulation results have demonstrated that the proposed algorithm offers superior performance to an existing frequency-based PEVD algorithm, with the advantage of not requiring a priori information regarding the paraunitary filter length.

When designing PEVD implementations for real applications, the algorithm described in this paper could be extremely useful, provided that  $K$  is not prohibitively large.

#### ACKNOWLEDGEMENTS

Fraser Coutts is the recipient of a Caledonian Scholarship; we would like to thank the Carnegie Trust for their support.

This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/K014307/1.

## REFERENCES

- [1] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix Polynomials*. Academic Press, New York, 1982.
- [2] S. Weiss, S. Bendoukha, A. Alzin, F. K. Coutts, I. K. Proudler, and J. Chambers. MVDR broadband beamforming using polynomial matrix techniques. In *EUSIPCO*, pp. 839–843, Nice, France, Sep. 2015.
- [3] A. Alzin, F. K. Coutts, J. Corr, S. Weiss, I. K. Proudler, and J. A. Chambers. Adaptive broadband beamforming with arbitrary array geometry. In *IET/EURASIP ISP*, London, UK, Dec. 2015.
- [4] M. Alrmah, S. Weiss, and S. Lambbotharan. An extension of the music algorithm to broadband scenarios using polynomial eigenvalue decomposition. In *19th European Signal Processing Conference*, pp. 629–633, Barcelona, Spain, Aug. 2011.
- [5] S. Weiss, M. Alrmah, S. Lambbotharan, J. McWhirter, and M. Kaveh. Broadband angle of arrival estimation methods in a polynomial matrix decomposition framework. In *IEEE 5th Int. Workshop Comp. Advances in Multi-Sensor Adaptive Proc.*, St. Martin, pp. 109–112, Dec. 2013.
- [6] F. K. Coutts, K. Thompson, S. Weiss, and I. K. Proudler. Impact of Fast-Converging PEVD Algorithms on Broadband AoA Estimation. In *SSPD*, London, UK, Dec. 2017.
- [7] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, Englewood Cliffs, 1993.
- [8] J. G. McWhirter, P. D. Baxter, T. Cooper, S. Redif, and J. Foster. An EVD Algorithm for Para-Hermitian Polynomial Matrices. *IEEE TSP*, 55(5):2158–2169, May 2007.
- [9] S. Icart, P. Comon. Some properties of Laurent polynomial matrices. In *IMA Int. Conf. Math. Signal Proc.*, Birmingham, UK, Dec. 2012.
- [10] S. Weiss, J. Pestana, and I. K. Proudler. On the Existence and Uniqueness of the Eigenvalue Decomposition of a Parahermitian Matrix. *IEEE TSP*, accepted for publication, 2018.
- [11] S. Redif, S. Weiss, and J. McWhirter. Sequential matrix diagonalization algorithms for polynomial EVD of parahermitian matrices. *IEEE TSP*, 63(1):81–89, Jan. 2015.
- [12] J. Corr, K. Thompson, S. Weiss, J. McWhirter, S. Redif, and I. K. Proudler. Multiple shift maximum element sequential matrix diagonalisation for parahermitian matrices. In *IEEE Workshop on Statistical Signal Processing*, pp. 312–315, Gold Coast, Australia, June 2014.
- [13] F. K. Coutts, J. Corr, K. Thompson, I. K. Proudler, and S. Weiss. Divide-and-Conquer Sequential Matrix Diagonalisation for Parahermitian Matrices. In *SSPD*, London, UK, Dec. 2017.
- [14] F. K. Coutts, K. Thompson, I. K. Proudler, and S. Weiss. Restricted Update Sequential Matrix Diagonalisation for Parahermitian Matrices. In *IEEE CAMSAP*, Curacao, Dec. 2017.
- [15] P. Vaidyanathan. Theory of optimal orthonormal subband coders. *IEEE TSP*, 46(6):1528–1543, June 1998.
- [16] M. Tohidian, H. Amindavar, and A. M. Reza. A DFT-based approximate eigenvalue and singular value decomposition of polynomial matrices. *EURASIP J. Adv. Signal Process.*, 2013:93, December 2013.
- [17] F. K. Coutts, K. Thompson, I. K. Proudler, and S. Weiss. A Comparison of Iterative and DFT-Based Polynomial Matrix Eigenvalue Decompositions. In *IEEE CAMSAP*, Curacao, Dec. 2017.
- [18] M. J. D. Powell. A new algorithm for unconstrained optimization. *Nonlinear programming*, pp. 31–65, 1970.
- [19] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, New York, 1999.